

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) A method of producing an executable file for execution by a computer, comprising ~~the steps of~~:

receiving a plurality of programming language statements comprising a source program into the computer, wherein the language statements include declarations defining attribute information for a symbol reference or symbol definition;

translating the source program into an object module, wherein the object module is capable of including: a symbol reference; a symbol definition; attribute information for the symbol reference derived from the language statements declaring the attribute information; and attribute information for the symbol definition derived from the language statements declaring the attribute information;

BI binding object modules into a program object, wherein the attribute information is available when binding object modules into the program object; and

resolving in the program object an external symbol reference in the object module with an external symbol definition in another object module.

2. (Original) The method of claim 1, wherein the language statement is capable of indirectly declaring extended attribute information defined in another location in the object module.

3. (Original) The method of claim 1, wherein the object module is further capable of including fixed attribute information derived from language statements declaring attribute information for the symbol reference and symbol definition.

4. (Original) The method of claim 1, wherein the attribute information is extended attribute information.

5. (Original) The method of claim 1, wherein the object module is further capable of including an address constant for a symbol referenced in the module and attribute information derived from language statements declaring attribute information for the address constant.

6. (Original) The method of claim 5, wherein the object module is further capable of including additional address constants for additional references to the symbol reference in the object module and different attribute information sets for the address constants for the additional references.

B1
7. (Original) The method of claim 6, wherein resolving the symbol reference and definition further comprises performing a compatibility check using the attribute information, wherein a separate compatibility check is performed for each reference to a symbol for which there is a separate address constant and separate attribute information for each address constant.

8. (Original) The method of claim 1, wherein the object module further includes an External Symbol Directory (ESD) including a record capable of indicating a symbol in the program, a location of the symbol in the program, and a pointer to attribute information in the program for the symbol.

9. (Original) The method of claim 1, wherein the object module further includes a Relocation List Directory (RLD) including a record capable of indicating the location of the referenced symbol, a location of an address constant for the referenced symbol in the program, and a pointer to attribute information for the address constant in the program.

10. (Original) The method of claim 1, wherein resolving the symbol reference and definition further comprises performing a compatibility check using signature data for the symbol definition and symbol reference.

11. (Original) The method of claim 10, further comprising determining whether the compatibility check failed and processing the attribute information declared for the symbol reference and definition that failed the compatibility check to determine a cause of the incompatibility.

12. (Currently Amended) A system for producing an executable file, comprising:
a computer;
program logic executed by the computer, comprising:

(i) means for receiving a plurality of programming language statements comprising a source program into the computer, wherein the language statements include declarations defining attribute information for a symbol reference or a symbol definition;

(ii) means for translating the source program into an object module, wherein the object module is capable of including: a symbol reference; a symbol definition; attribute information for the symbol reference derived from the language statements declaring the attribute information; and attribute information for the symbol definition derived from the language statements declaring the attribute information;

(iii) binding object modules into a program object, wherein the attribute information is available when binding object modules into the program object; and

(iv) means for resolving in the program object an external symbol reference in the object module with an external symbol definition in another object module.

13. (Original) The system of claim 12, wherein the language statement is capable of indirectly declaring extended attribute information defined in another location in the object module.

14. (Original) The system of claim 12, wherein the object module is further capable of including fixed attribute information derived from language statements declaring attribute information for the symbol reference and symbol definition.

15. (Original) The system of claim 12, wherein the attribute information is extended attribute information.

16. (Original) The system of claim 12, wherein the object module is further capable of including an address constant for a symbol referenced in the module and attribute information derived from language statements declaring attribute information for the address constant.

31
17. (Original) The system of claim 16, wherein the object module is further capable of including additional address constants for additional references to the symbol reference in the object module and different attribute information sets for the address constants for the additional references.

18. (Original) The system of claim 17, wherein resolving the symbol reference and definition further comprises performing a compatibility check using the attribute information, wherein a separate compatibility check is performed for each reference to a symbol for which there is a separate address constant and separate attribute information for each address constant.

19. (Original) The system of claim 12, wherein the object module further includes an External Symbol Directory (ESD) including a record capable of indicating a symbol in the program, a location of the symbol in the program, and a pointer to attribute information in the program for the symbol.

20. (Original) The system of claim 12, wherein the object module further includes a Relocation List Directory (RLD) including a record capable of indicating the location of the

referenced symbol, a location of an address constant for the referenced symbol in the program, and a pointer to attribute information for the address constant in the program.

21. (Original) The system of claim 12, wherein resolving the symbol reference and definition further comprises performing a compatibility check using signature data for the symbol definition and symbol reference.

22. (Original) The system of claim 21, further comprising determining whether the compatibility check failed and processing the attribute information declared for the symbol reference and definition that failed the compatibility check to determine a cause of the incompatibility.

23. (Currently Amended) An article of manufacture for producing an executable file for execution by a computer, the article of manufacture comprising computer usable media including at least one computer program embedded therein that is capable of causing a computer to perform:

receiving a plurality of programming language statements comprising a source program into the computer, wherein the language statements include declarations defining attribute information for a symbol reference or a symbol definition;

translating the source program into an object module, wherein the object module is capable of including: a symbol reference; a symbol definition; attribute information for the symbol reference derived from the language statements declaring the attribute information; and attribute information for the symbol definition derived from the language statements declaring the attribute information;

binding object modules into a program object, wherein the attribute information is available when binding object modules into the program object; and

resolving in the program object an external symbol reference in the object module with an external symbol definition in another object module.

24. (Original) The article of manufacture of claim 23, wherein the language statement is capable of indirectly declaring extended attribute information defined in another location in the object module.

25. (Original) The article of manufacture of claim 23, wherein the object module is further capable of including fixed attribute information derived from language statements declaring attribute information for the symbol reference and symbol definition.

26. (Original) The article of manufacture of claim 23, wherein the attribute information is extended attribute information.

B1
27. (Original) The article of manufacture of claim 23, wherein the object module is further capable of including an address constant for a symbol referenced in the module and attribute information derived from language statements declaring attribute information for the address constant.

28. (Original) The article of manufacture of claim 27, wherein the object module is further capable of including additional address constants for additional references to the symbol reference in the object module and different attribute information sets for the address constants for the additional references.

29. (Original) The article of manufacture of claim 28, wherein resolving the symbol reference and definition further comprises performing a compatibility check using the attribute information, wherein a separate compatibility check is performed for each reference to a symbol for which there is a separate address constant and separate attribute information for each address constant.

30. (Original) The article of manufacture of claim 23, wherein the object module further includes an External Symbol Directory (ESD) including a record capable of indicating a symbol in the

program, a location of the symbol in the program, and a pointer to attribute information in the program for the symbol.

31. (Original) The article of manufacture of claim 23, wherein the object module further includes a Relocation List Directory (RLD) including a record capable of indicating the location of the referenced symbol, a location of an address constant for the referenced symbol in the program, and a pointer to attribute information for the address constant in the program.

32. (Original) The article of manufacture of claim 23, wherein resolving the symbol reference and definition further comprises performing a compatibility check using signature data for the symbol definition and symbol reference.

B1
33. (Original) The article of manufacture of claim 32, further comprising determining whether the compatibility check failed and processing the attribute information declared for the symbol reference and definition that failed the compatibility check to determine a cause of the incompatibility.

34. (New) The method of claim 1, wherein the attribute information describes whether the symbol definition or symbol reference is to data or executable code.

35. (New) The method of claim 34, wherein the attribute information indicates data types and structures if the symbol definition or symbol reference is for data and indicates a number and types of arguments passed or received if the symbol definition or symbol reference is for executable code.

36. (New) The system of claim 12, wherein the attribute information describes whether the symbol definition or symbol reference is to data or executable code.

37. (New) The system of claim 36, wherein the attribute information indicates data types and structures if the symbol definition or symbol reference is for data and indicates a number and types of arguments passed or received if the symbol definition or symbol reference is for executable code.

B1
38. (New) The article of manufacture of claim 23, wherein the attribute information describes whether the symbol definition or symbol reference is to data or executable code.

39. (New) The article of manufacture of claim 38, wherein the attribute information indicates data types and structures if the symbol definition or symbol reference is for data and indicates a number and types of arguments passed or received if the symbol definition or symbol reference is for executable code.
